Homework 1

CMPT333N
Due on February 7th, 2020

Truth Tables

Problem 1

Give the rule for computing the

a) NAND

b) NOR

c) $\equiv$

of two columns of a truth table.

Problem 2

Compute the truth table for the following expressions and their subexpressions.

a) $(p \to q) \equiv (\texttt{NOT } p \texttt{ OR } q)$

b) $p \to (q \to (r \texttt{ OR NOT } p))$

c) $(p \texttt{ OR } q) \to (p \texttt{ AND } q)$

Problem 3

To what set operator does the logical expression p AND NOT q correspond?

Problem 4

Give examples to show that $\to$, NAND, and NOR are not associative.

Problem 5

A Boolean function f *does not depend on the first argument* if

$$f(\texttt{TRUE}, x_2, x_3, ..., x_k) = f(\texttt{FALSE}, x_2, x_3, ..., x_k)$$

for any truth values $x_2, x_3, ..., x_k$. Similarly, we can say f does not depend on its ith argument if the value of f never changes when its ith argument is switched between TRUE and FALSE. How many Boolean functions of two arguments do not depend on their first or second argument (or both)?

Problem 6

Construct truth tables for the 16 Boolean functions of two variables. How many of these functions are commutative?

Problem 7

The binary *exclusive−or* function, $\otimes$, is defined to have value `TRUE` if and only if exactly one of its arguments are `TRUE`.

a) Draw the truth table for $\otimes$.

b) Is $\otimes$ commutative? Is it associative?

From Boolean Functions to Logical Expressions

Problem 8

The truth table below defines two Boolean functions, a and b, in terms of variables p, q, and r. Write *sum-of-products* expressions for each of these functions.

p	q	r	a	b
0	0	0	0	1
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Problem 9

Write *product-of-sums* expressions for

a) Function a of the previous table.

b) Function b of the previous table.

c) Function z of the table below.

x	y	c	d	z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Designing Logical Expressions by Karnaugh Maps

Problem 10

Draw the Karnaugh maps for the following functions of variables p, q, r, and s.

a) The function that is `TRUE` if one, two, or three of p, q, r, and s are `TRUE`, but not if zero or all four are `TRUE`.

b) The function that is `TRUE` if up to two of p, q, r, and s are `TRUE`, but not if three or four are `TRUE`.

c) The function that is `TRUE` if one, three, or four of p, q, r, and s are `TRUE`, but not if zero or two are `TRUE`.

d) The function represented by the logical expression $pqr \to s$.

e) The function that is `TRUE` if $pqrs$, regarded as a binary number, has value less than ten.

Problem 11

Find the implicants - other than the minterms - for each of your Karnaugh maps from Problem 10. Which of them are prime implicants? For each function, find a sum of prime implicants that covers all the 1's of the map. Do you need to use all the prime implicants?

Problem 12

Show that every product in a sum-of-products expression for a Boolean function is an implicant of that function.

Problem 13

One can also construct a product-of-sums expression from a Karnaugh map. We begin by finding rectangles of the types that form implicants, but with all points 0, instead of all points 1. Call such a rectangle an "anti-implicant." We can construct for each anti-implicant a sum of literals that is 1 on all points but those of the anti-implicant. For each variable x, this sum has literal x if the anti-implicant includes only points for which $x = 0$, and it has literal $\overline{x}$ if the anti-implicant has only points for which $x = 1$. Otherwise, the sum does not have a literal involving x. Find all the prime anti-implicants for your Karnaugh maps of Problem 10.

Problem 14

Using your answer to Problem 13, write product-of-sums expressions for each of the functions of Problem 10. Include as few sums as you can.

Problem 15

How many (a) 1×2 (b) 2×2 (c) 1×4 (d) 2×4 rectangles that form implicants are there in a 4×4 Karnaugh map? Describe their implicants as products of literals, assuming the variables are p, q, r, and s.

Tautologies

Problem 16

Which of the following expressions are tautologies?

a) $pqr \to p + q$

b) $((p \to q)(q \to r)) \to (p \to r)$

c) $(p \to q) \to p$

d) $(p \equiv (q + r)) \to (q \to pr)$

Problem 17

Suppose we had an algorithm to solve the tautology problem for a logical expression. Show how this algorithm could be used to

a) Determine whether two expressions were equivalent.

b) Solve the satisfiability problem

Some Algebraic Laws for Logical Expressions

1. $Reflexivity\ of\ equivalence : p \equiv p.$

2. $Commutative\ law\ for\ equivalence : (p \equiv q) \equiv (q \equiv p).$

3. $Transitive\ law\ for\ equivalence : ((p \equiv q)\ \texttt{AND}\ (q \equiv r)) \rightarrow (p \equiv r).$

4. $Equivalence\ of\ the\ negations : (p \equiv q) \equiv (\overline{p} \equiv \overline{q}).$

5. $The\ commutative\ law\ for\ \texttt{AND} : pq \equiv qp.$

6. $The\ associative\ law\ for\ \texttt{AND} : p(qr) \equiv (pq)r.$

7. $The\ commutative\ law\ for\ \texttt{OR} : (p + q) \equiv (q + p).$

8. $The\ associative\ law\ for\ \texttt{OR} : (p + (q + r)) \equiv ((p + q) + r).$

9. $The\ distributive\ law\ of\ \texttt{AND}\ over\ \texttt{OR} : p(q + r) \equiv (pq + pr).$

10. $1(\texttt{TRUE})\ is\ the\ identity\ for\ \texttt{AND} : (p\ \texttt{AND}\ 1) \equiv p.$

11. $0(\texttt{FALSE})\ is\ the\ identity\ for\ \texttt{OR} : p\ \texttt{OR}\ 0 \equiv p.$

12. $0\ is\ the\ annihilator\ for\ \texttt{AND} : (p\ \texttt{AND}\ 0) \equiv 0.$

13. $Elimination\ of\ double\ negations : (\texttt{NOT NOT}\ p) \equiv p.$

14. $The\ distributive\ law\ for\ \texttt{OR}\ over\ \texttt{AND} : (p + qr) \equiv ((p + q)(p + r)).$

15. $1\ is\ the\ annihilator\ for\ \texttt{OR} : (1\ \texttt{OR}\ p) \equiv 1.$

16. $Idempotence\ of\ \texttt{AND} : pp \equiv p.$

17. $Idempotence\ of\ \texttt{OR} : p + p \equiv p.$

18. $Subsumption.$
    (a) $(p + pq) \equiv p.$
    (b) $p(p + q) \equiv p.$

19. $Elimination\ of\ certain\ negations.$
    (a) $p(\overline{p} + q) \equiv pq.$
    (b) $p + \overline{p}q \equiv p + q.$

20. $DeMorgan's\ laws.$
    (a) $\texttt{NOT}\ (pq) \equiv \overline{p} + \overline{q}.$
    (b) $\texttt{NOT}\ (p + q) \equiv \overline{pq}.$
    (c) $(\texttt{NOT}\ (p_1p_2...p_k)) \equiv (\overline{p_1} + \overline{p_2} + ... + \overline{p_k}).$
    (d) $(\texttt{NOT}\ (p_1 + p_2 + ... + pk)) \equiv (\overline{p_1 p_2}...\overline{p_k}).$

21. $((p \to q)$ AND $(q \to p)) \equiv (p \equiv q)$.

22. $(p \equiv q) \to (p \to q)$.

23. *Transitivity of implication* : $((p \to q)$ AND $(q \to r)) \to (p \to r)$.

24. Implication with AND and OR:

    (a) $(p \to q) \equiv (\overline{p} + q)$.

    (b) $(p_1 p_2 ... p_n \to q) \equiv (\overline{p_1} + \overline{p_2} + ... + \overline{p_n} + q)$.

Problem 18

Check, by constructing the truth tables, that each of the laws 1 to 24 are tautologies.

Problem 19

We can substitute expressions for any propositional variable in a tautology and get another tautology. Substitute $x + y$ for p, yz for q, and $\bar{(}x)$ for r in each of the tautologies 1 to 24, to get new tautologies. Do not forget to put parentheses around the substituted expressions if needed.

Problem 20

Use laws given in this section to transform the first of each pair of expressions into the second. To save effort, you may omit steps that use laws 5 through 13, which are analogous to arithmetic. For example, commutative and associativity of AND and OR may be assumed.

a) Transform $pq + rs$ into $(p + r)(p + s)(q + r)(q + s)$.

b) Transform $pq + p\overline{q}r$ into $p(q + r)$.

Problem 21

Show that the subsumption laws, 18(a) and (b), follow from previously given laws, in the sense that it is possible to transform $p + pq$ into p and transform $p(p + q)$ into p using only laws 1 through 17.

Problem 22

Apply DeMorgan's laws to turn the following expressions into expressions where the only NOT's are applied to propositional variables (i.e., the NOT's appear in literals only).

1. NOT $(pq + \overline{pr})$

2. NOT (NOT $p + q($NOT $(r + \overline{s})))$

Problem 23

Simplify the following by using the subsumption laws and the commutative and associative laws for AND and OR.

1. $w\overline{x} + w\overline{x}y + \overline{z}\overline{x}w$

2. $(w + \overline{x})(w + y + \overline{z})(\overline{w} + \overline{x} + \overline{y})(\overline{x})$